Introduction:

Robust and High Performance Tools for Scientific Computing

Tony Drummond
Lawrence Berkeley National Laboratory
ACTS Collection Workshop
September 4-7, 2002

Motivation

Grand Challenges are ..fundamental problems in science and engineering, with potentially broad social, political, and scientific impact, that could be advanced by applying high performance computer resources

Office of Science and Technology

- Some grand challenges: electronic structure of materials, turbulence, genome sequencing and structural biology, global climate modeling, speech and language studies, pharmaceutical design, pollution, etc. .

Motivation

With the development of new kinds of equipment of greater capacity, and particularly of greater speed, it is almost certain that new methods will have to be developed in order to make the fullest use of this equipment. It is necessary not only to design machines for the mathematics, but also to develop a new mathematics for the machines - 1952, Hartree

- **Metropolis** grew out of physical chemistry in 1950's through attempts to calculate statistical properties of chemical reactions. Some areas of application: astrophysics, many areas engineering, and chemistry
- **Fast Fourier Transform** (**FFT**): implementation of Fourier Analysis. Some areas of application: image and signal processing, seismology, physics, radiology, acoustics and engineering
- **Multigrids**: method for solving a wide variety of PDE. Some areas of application: physics, biophysics and engineering

Partial Matrix of Methods and Disciplines

	Climate Change	Material Science	High Enregy Physics	Astrophysics Cosmology	Biology	Chemistry	Fusion
Monte Carlo (Quantum and Classical)	PCM CCSM POP	Quantum MC Classical KMC	FASTER SYNPOL	FASTER SYNPOL		NWChem	
Fast Fourier Transform		VASP Paratec Petot Escan	IMPACT LANGEVIN3D MAD9P ccSHT		SPIDER NAMD	NAMD	WARP GTC
Fast Multipole & Variants		Classical MD	IMPACT LANGEVIN3D QuickPIC		Classical MD	NWChem Classical MD	
Sparse Linear systems	PCM CCSM POP	O(N) Methods	OMEGA3P		SPIDER	pVarDen	NIMROD
Eigenvalue Solvers		DFT FLAPW PW codes	OMEGA3P		DFT SPIDER	NWChem Gaussian QChem	
Dense Linear Solvers		LSMS FLAPW		MADCAP		NWChem Gaussian	GTC
Adaptive Mesh Refinement	BoxLib Paramesh		BoxLib Paramesh	FLASH Paramesh		pVarDen BoxLib	WARP BOX Chombo

Motivation

<u>Computational science</u>: can be characterized by the needs to gain understanding through the analysis of mathematical models using high performing performing computers

Community

- Scientists
- Engineers
- Mathematicians
- Economists, artists

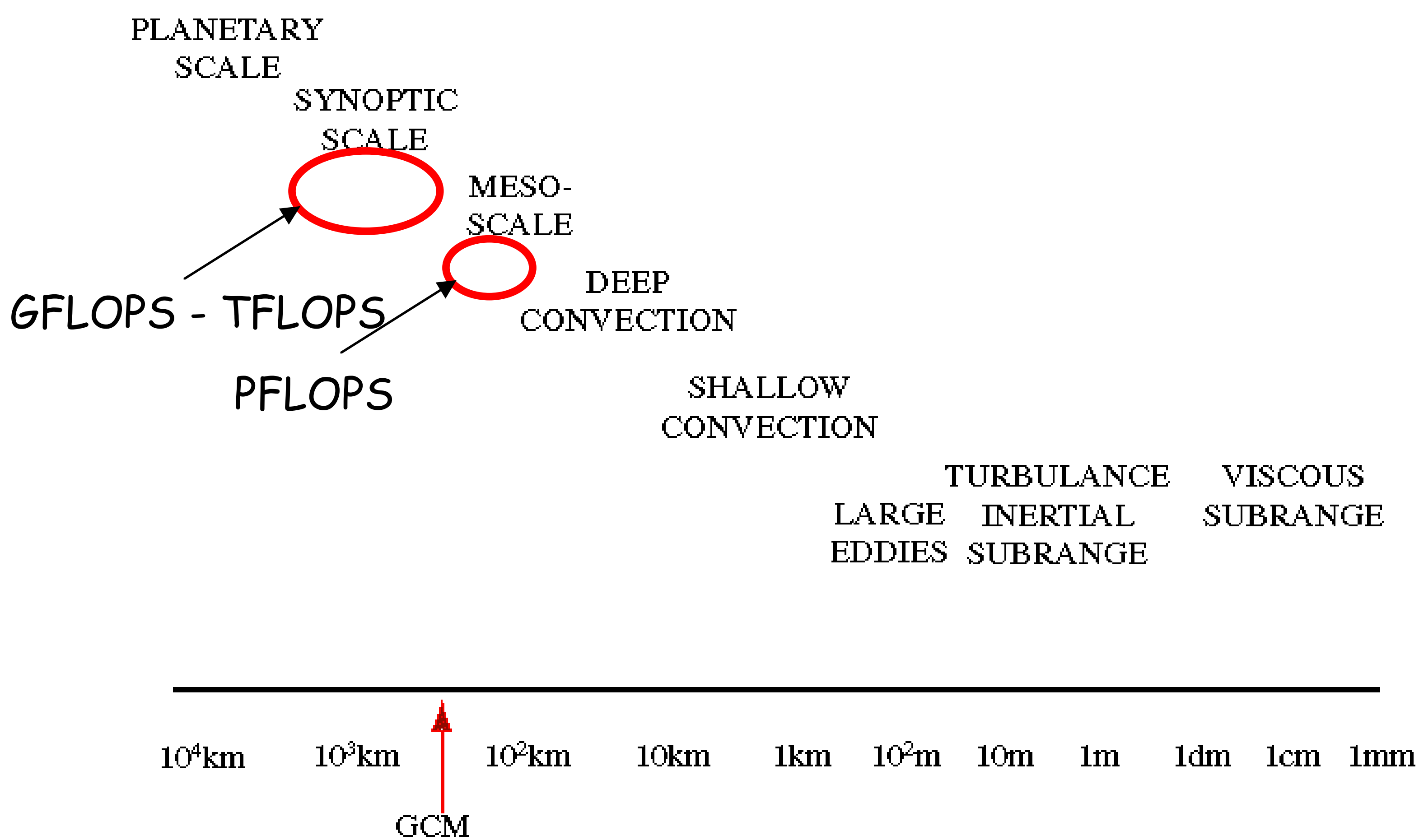
Multidisciplinary!

<u>Computer Science</u> Provides services ranging from Networking and visualization tools to algorithms	<u>Mathematics</u>: Credibility of algorithms (error analysis, exact solutions, expansions, uniqueness proofs and theorems)

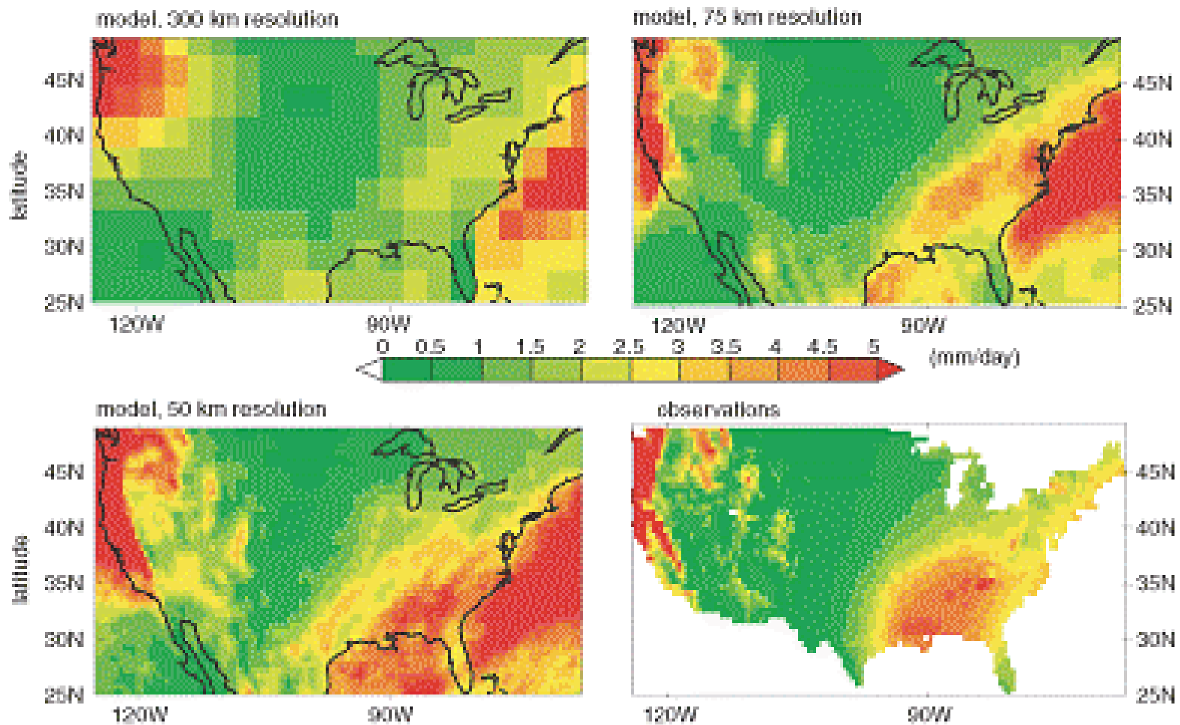
Some lessons learned from Earth System Modeling

Motivation - Example I

SPECTRUM OF ATMOSPHERIC PHENOMENA



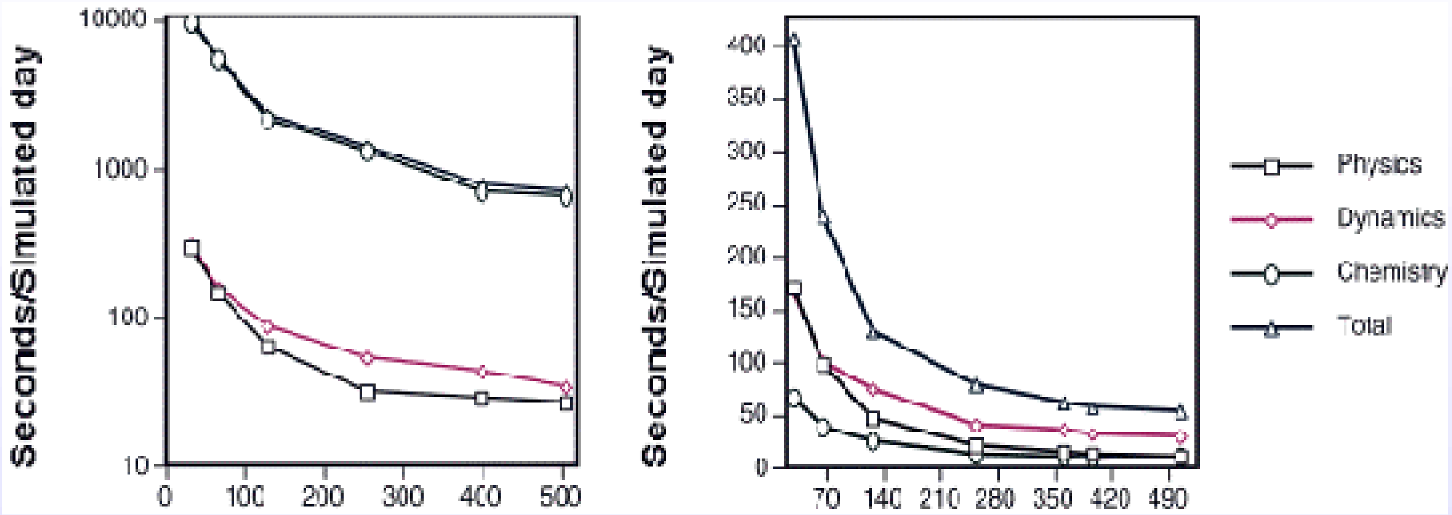
Motivation - Example I



Duffy et. al.,
Lawrence Livermore National Laboratory

- CCM3 - spectral truncations of T170 and T239
- 50 Km spatial resolution is 32 times more grid cells and takes roughly **200 times longer** to run

Motivation - Example II



AGCM/ACM
2.5 long x 2 lat, 30 layers
25-chemical species

AGCM/ACM
2.5 long x 2 lat, 30 layers
2-chemical species

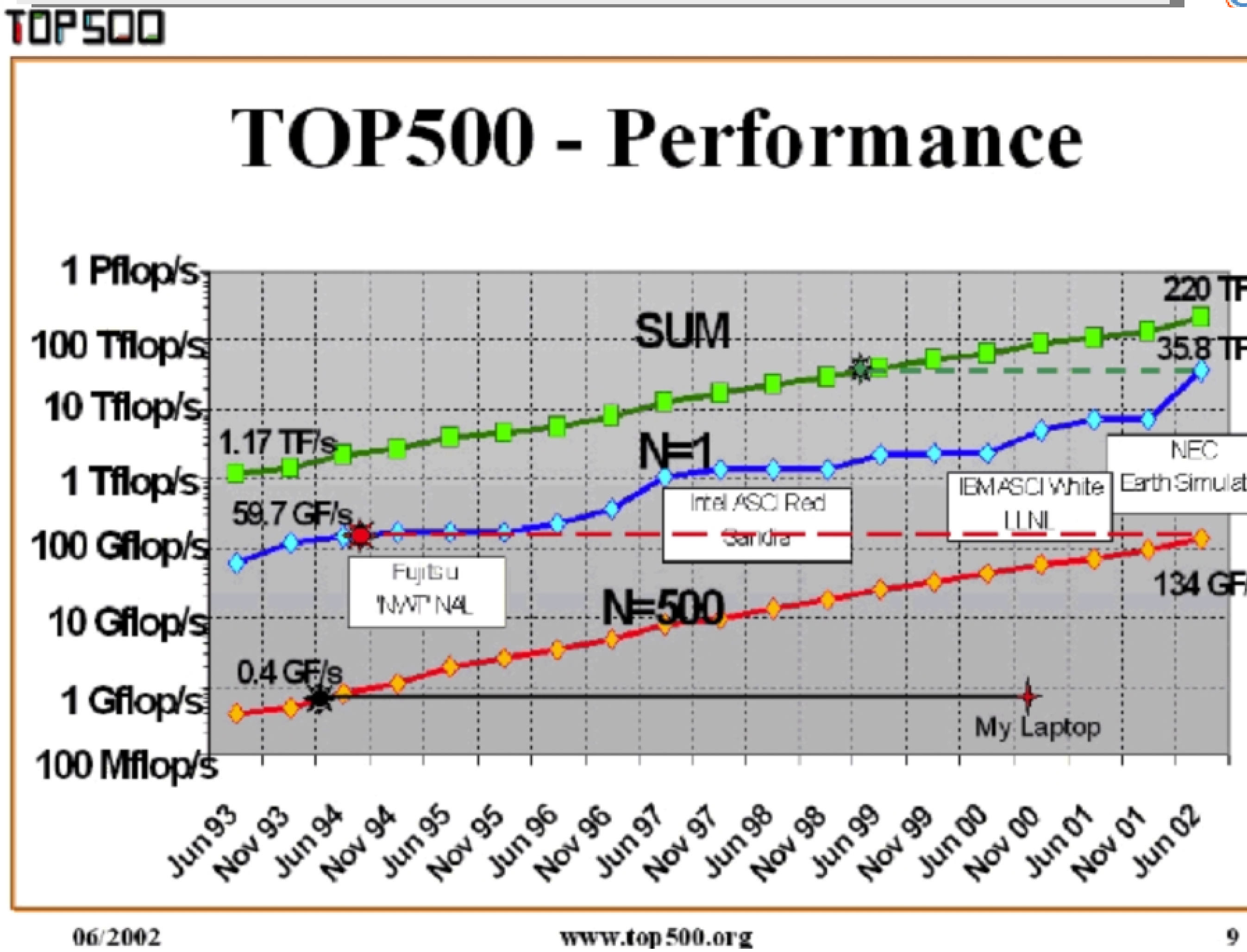
- **Non-linear demand for resources (CPU - Memory)**
- **Multi-disciplinary application is more demanding**

The Hardware

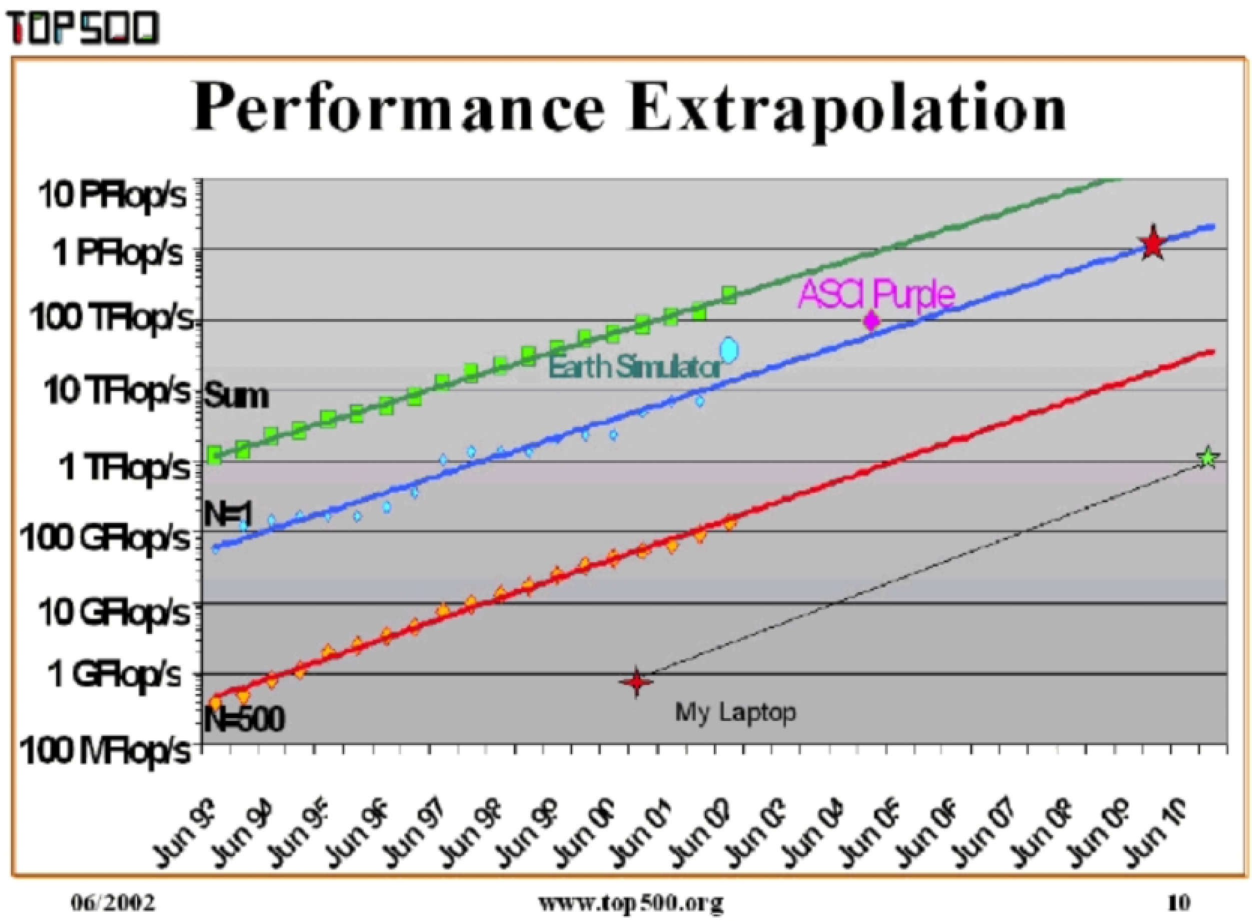
TOP 500 – June 2002

Rank	Manufacturer	Computer	Rmax	Installation Site	Country	Year	Area of Installation	# Proc	Rpeak	Nmax	N1/2
1	NEC	Earth-Simulator	**35860**	Earth Simulator Center Kanazawa	Japan	2002	Research	5120	40960	1075200	266240
2	IBM	ASCI White, SP Power3 375 MHz	**7226**	Lawrence Livermore National Laboratory Livermore	USA	2000	Research Energy	8192	12288	518096	179000
3	Hewlett-Packard	AlphaServer SC ES45/1 GHz	**4463**	Pittsburgh Supercomputing Center Pittsburgh	USA	2001	Academic	3016	6032	280000	85000
4	Hewlett-Packard	AlphaServer SC ES45/1 GHz	**3980**	Commissariat a l'Energie Atomique (CEA) Bruyeres-le-Chatel	France	2001	Research	2560	5120	360000	85000
5	IBM	SP Power3 375 MHz 16 way	**3052**	NERSC/LBNL Berkeley	USA	2001	Research	3328	4992	371712	102400
6	Hewlett-Packard	AlphaServer SC ES45/1 GHz	**2916**	Los Alamos National Laboratory Los Alamos	USA	2002	Research	2048	4096	272000	.
7	Intel	ASCI Red	**2379**	Sandia National Laboratories Albuquerque	USA	1999	Research	9632	3207	362880	75400
8	IBM	pSeries 690 Turbo 1.3GHz	**2310**	Oak Ridge National Laboratory Oak Ridge	USA	2002	Research	864	4493	275000	62000
9	IBM	ASCI Blue-Pacific SST, IBM SP 604e	**2144**	Lawrence Livermore National Laboratory Livermore	USA	1999	Research Energy	5808	3868	431344	.
10	IBM	pSeries 690 Turbo 1.3GHz	**2002**	IBM/US Army Research Laboratory (ARL) Poughkeepsie	USA	2002	Vendor	768	3994	252000	.
		SP Power3 375 MHz		Atomic Weapons							

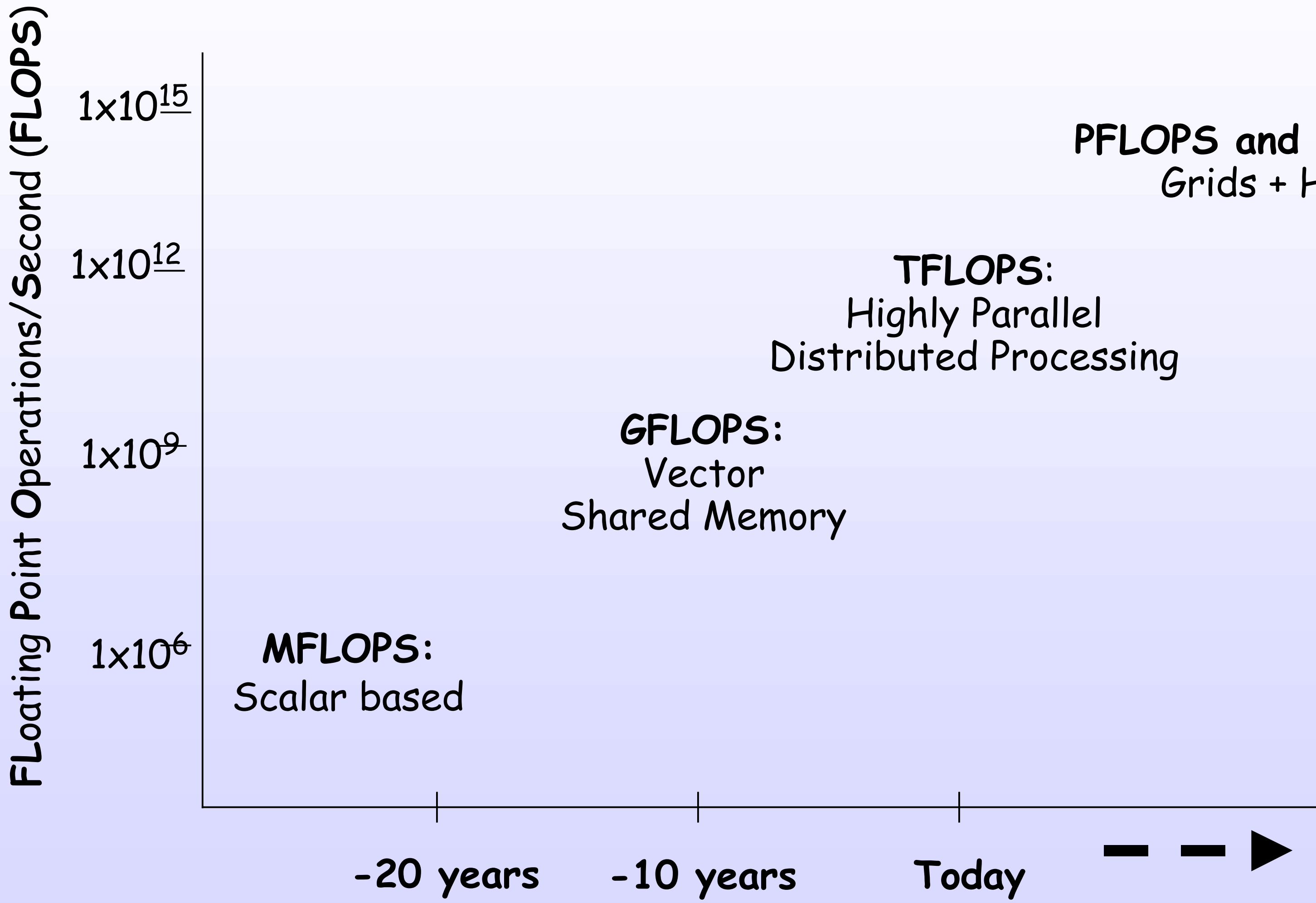
Available Hardware for High Performance Computing



Hardware for High Performance Computing



Evolution of High Performance Computers



The GRID

- A large pool of resources
    - Computers
    - Networks
    - Software
    - Databases
    - Instruments
    - people

<u>Requirements from GRID implementation:</u>

- Ubiquitous: ability to interface to the grid at any point and leverage whatever is available
- Resource Aware: manage heterogeneity of resources
- Adaptive: tailored to obtain maximum performance from resources

Using today's hardware to tackle today's Grand Challenges

Q. Why is it **still** difficult to obtain High Performance?

Some common and *interesting* answers

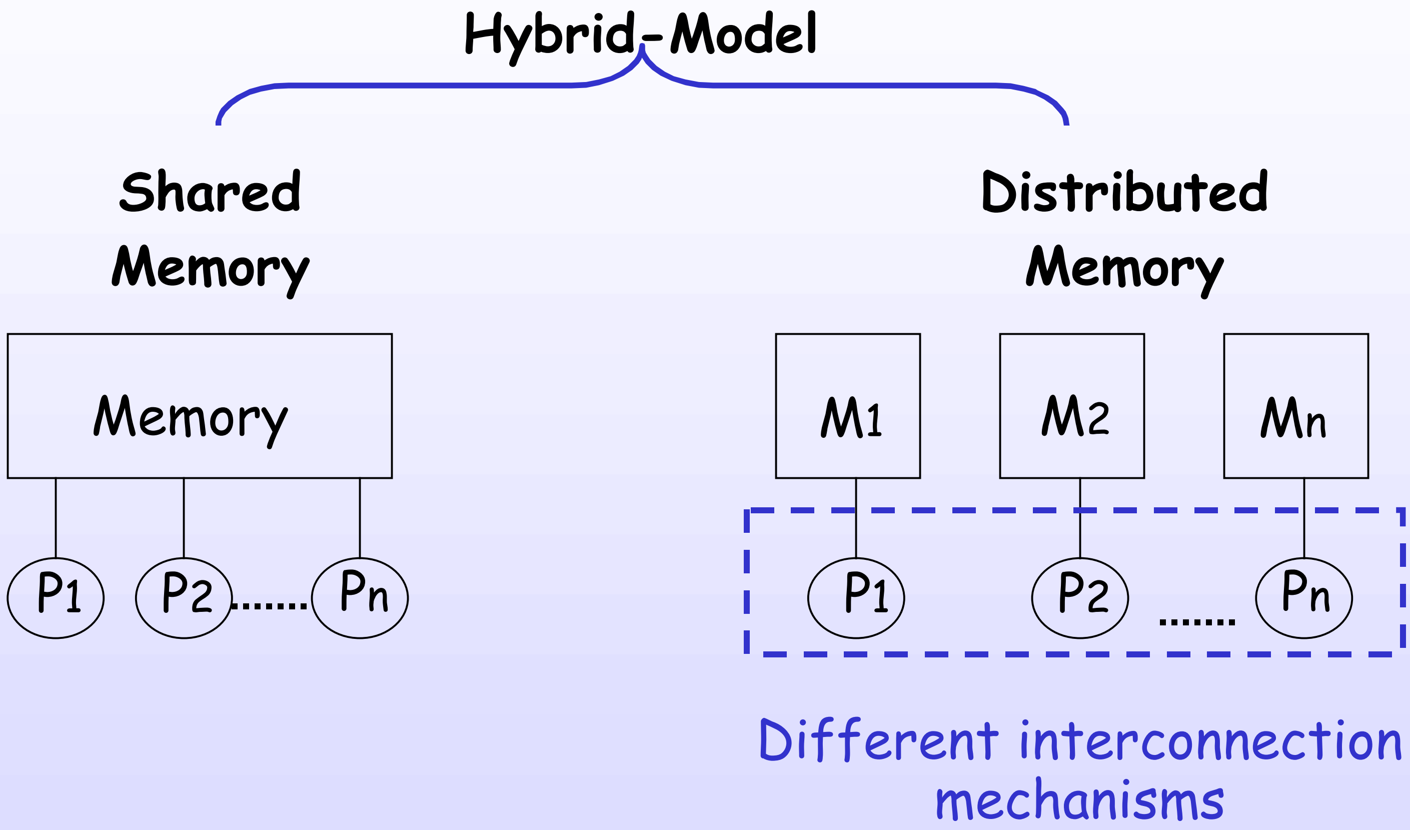
- Technology
- Memory latency
- Algorithms
- Programming Practices

⋮

Some options for New Architectures

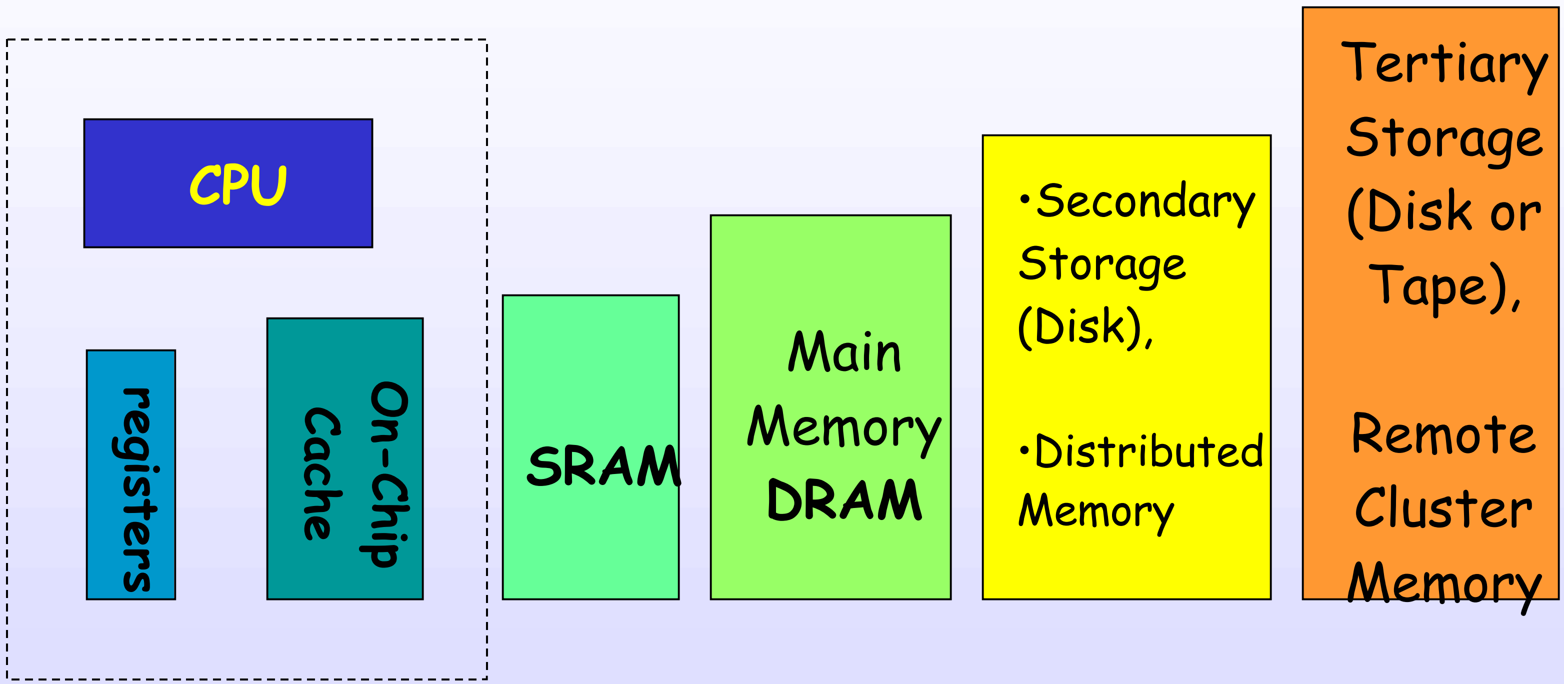
OPTION	SOFTWARE IMPACT	COST	TIMELINE
Modification of commodity processors	Minimal	2 or 3 times commodity?	Can be achieved in three years
U.S.-made vector architecture	Moderate	2 or 3 times commodity at present	Deliverable in 2003 and beyond
Processor-in-memory (Blue Gene/L)	Extensive	Unknown, 2 to 5 times commodity?	Only prototypes available now
Japanese- made vector architecture	Moderate	2.5 to 3 times commodity at present	Available now
Research Architectures (Streams, VIRAM..)	Extensive or unknown	Unknown	Academic research prototypes only available now

Memory Latency



Memory Hierarchy

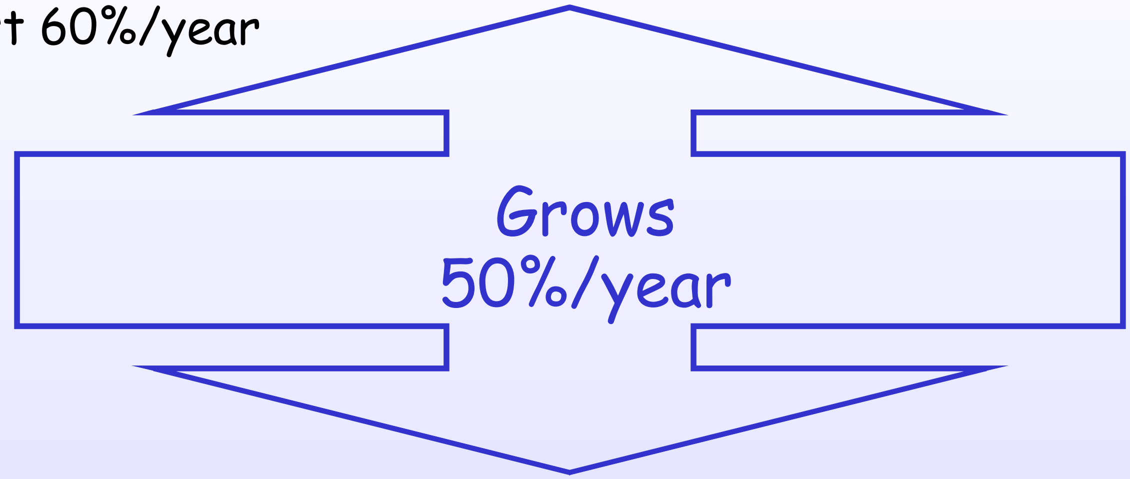
- *Where is the data? Why is data locality important?*



Speed	1's ns	10's ns		100's ns	.1's -10's ms	10's s
Size	100's bytes	Kbytes		Mbytes	Gbytes	Tbytes

CPU vs. DRAM Performance

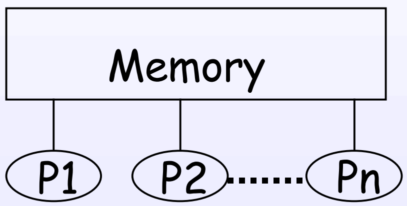
• Since 1980's, µProcs performance has increased at a rate of almost 60%/year



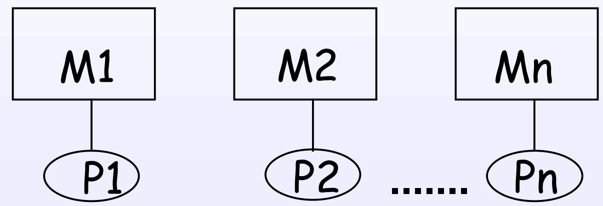
• Since 1980's, DRAM (latency) has improved at a rate of almost 9%/year

Parallel Programming Paradigms

Shared Memory



Distributed Memory



- Data parallelism

- easier to implement
- shared memory space
- mutual exclusion, contention

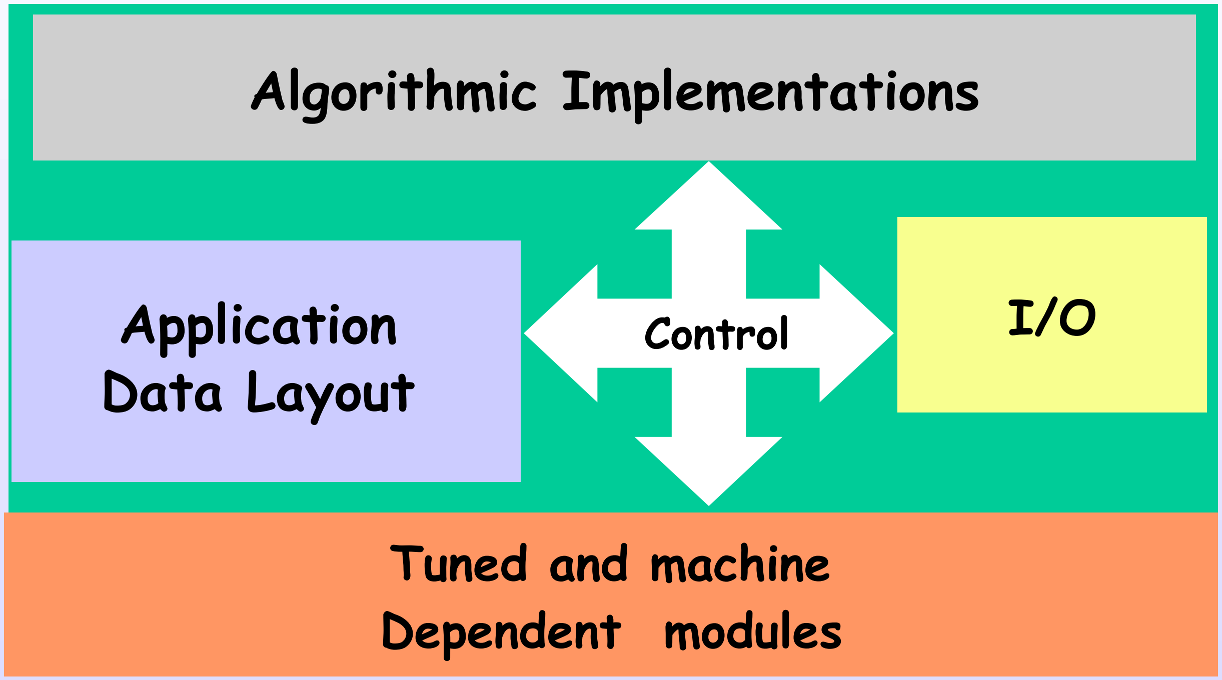
- virtual shared memory
- data is implicitly available to all

- Message Passing

- shared area is use for sending and receiving data

- Implicit mutual exclusion
- Only explicit synch
- Depends on Memory Hierarchy and Network

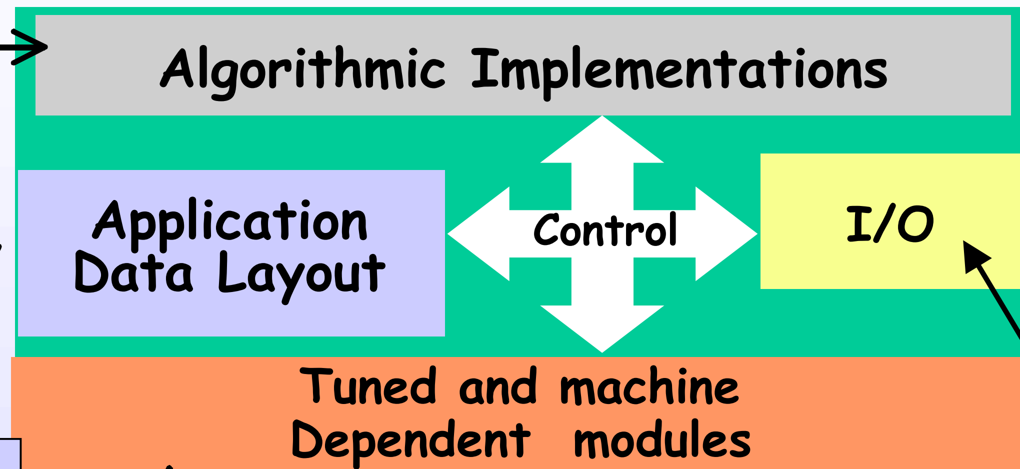
Large Scientific Codes: *A Common Programming Practice*



Shortcomings

New Architecture:
- May or may not need re-rewriting

New Developments:
- Difficult to compare



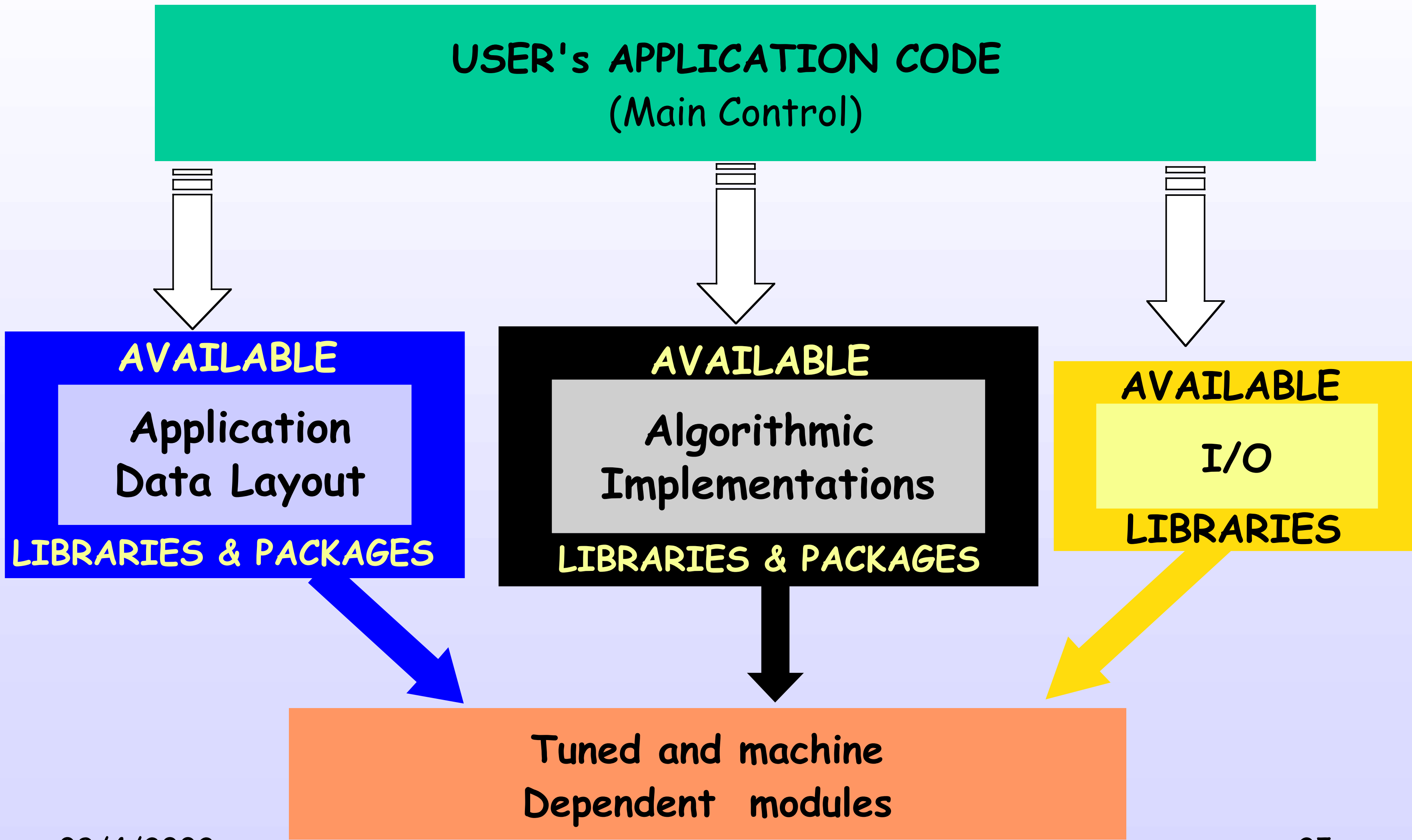
New Architecture:
- Extensive re-rewriting

New or extended Physics:
- Extensive re-rewriting or increase overhead

New Architecture or S/W
- Extensive tuning
- May require new programming paradigms
- **Difficult to maintained!**

New Architecture:
- Minimal to Extensive rewriting

Alternative Programming Approach



Software Development Levels of abstraction

- Scientific or engineering context
- Domain expertise

- Simulation codes
- Data Analysis codes

- Templates
- Scientific Computing Tools
- General Purpose Libraries

- Data Structures
- Algorithms
- Code Optimization
- Programming Languages
- O/S - Compilers

Hardware - Middleware - Firmware

What is the DOE ACTS Collection?

- Advanced CompuTational Software
- Tools for developing parallel applications
  - Developed (primarily) at DOE Labs
  - Separate projects originally
  - ~ 20 tools
- ACTS is an “umbrella” project
  - Leverage numerous independently funded projects
  - Collect tools in a toolkit

ACTS: *Project Goals*

- Extended support for *experimental software*
- Make ACTS tools available on DOE computers
- Provide technical support (*acts-support@nersc.gov*)
- Maintain ACTS information center (*http://acts.nersc.gov*)
- Coordinate efforts with other supercomputing centers
- Enable large scale scientific applications
- Educate and train

Related Activities

- **Software Repositories**:
  - Netlib: *http://www.netlib.org*
  - HPC-Netlib: *http://www.nhse.org/hpc-netlib*
  - National HPCC Software Exchange NHSE: *http://www.nhse.org*
  - Guide to Available Mathematical Software: *http://gams.nist.gov*
  - MGNet: *http://www.mgnet.org*
  - NEOS: *http://www-fp.mcs.anl.gov/otc/Guide*
  - OO Numerics: *http://oonumerics.org/oon*
- **Portable timing routines, tools for debugging, compiler technologies:**
  - Ptools: *http://www.ptools.org*
  - Center for Programming Models for Scalable Parallel Computing: *http://www.pmodels.org*
- **Education:**
  - Computational Science Educational Project: *http://csep1.phy.ornl.gov*
  - UCB's Applications of Parallel Computers: *http://www.cs.berkeley.edu/~demmel/cs267_Spr99*
  - MIT's Applied Parallel Computing: *http://www.mit.edu/~cly/18.337*
  - Dictionary of algorithms, data structures and related definitions: *http://www.nist.gov/dads*

Why is ACTS unique?

- Extended support for tools
- Accumulates the expertise and user feedback on the use of the software tools and scientific applications that used them:
  - independent software evaluations
  - participation in the developer user groups e-mail list
  - presentation of a gallery of applications
  - leverage between tool developers and tool users
  - workshops and tutorials
  - tool classification
  - support

What needs to be computed?

ScaLAPACK

Aztec/Trilinos

SuperLU

$$Ax = b$$

$$Az = \lambda z$$

$$A = U\Sigma V^T$$

$$\min\left\{\tfrac{1}{2}\|r(x)\|^2 : x_l \le x \le x_u\right\}$$

OPT++

PDEs

ODEs

TAO

SUNDIALS

PETSc

Hypre

What codes are being developed?

Global Arrays

PAWS

Overture

Parallel programs that use large distributed arrays

Coupling distributed applications

Support for Grids and meshes

Infrastructure for distributed computing

On-line visualization and computational stearing

Language Interoperability

Performance analysis and monitoring

Chasm

CUMULVS

Globus

TAU

How much effort is involved in using these tools?

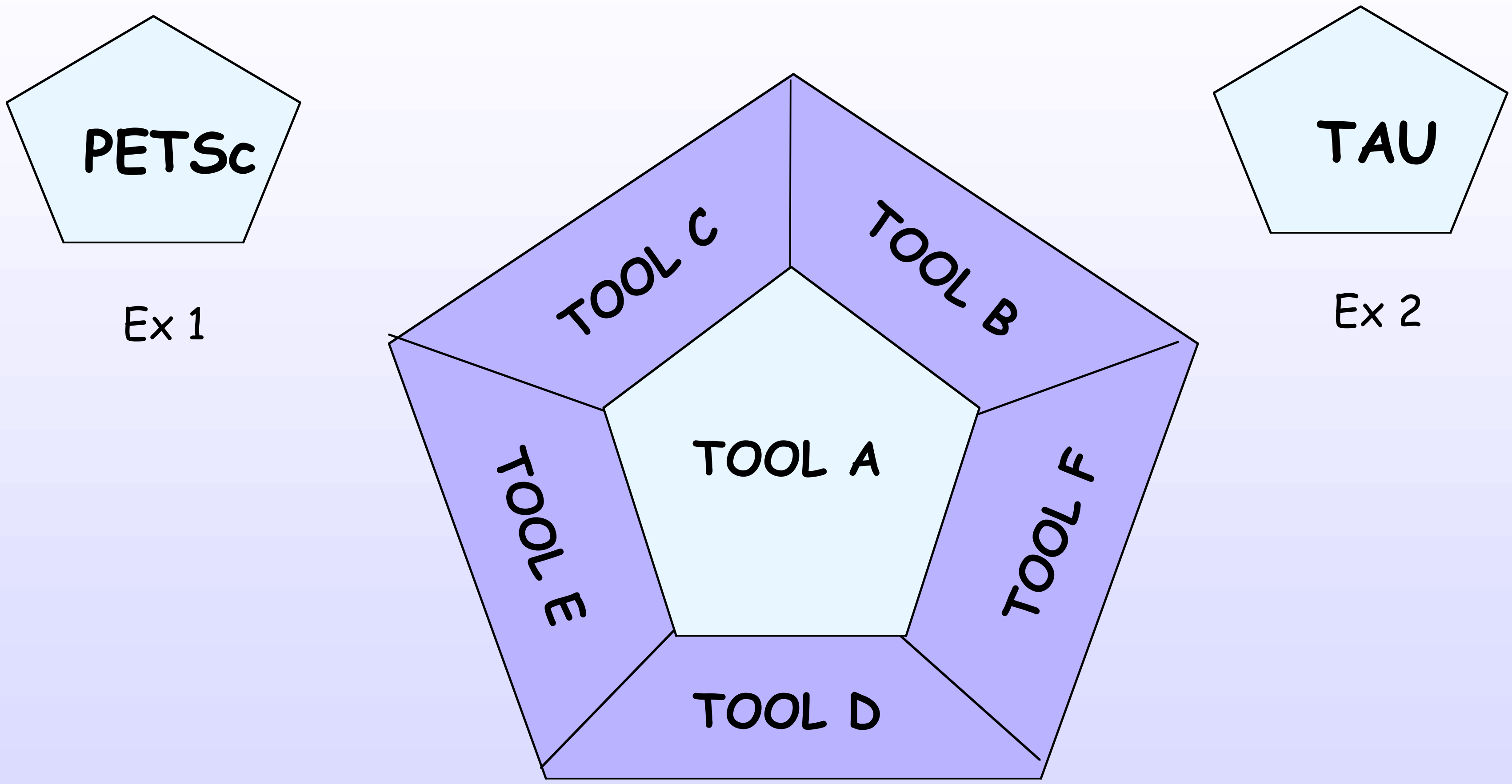
Using the ACTS Collection

- Must of the tools provide interfaces (calling functions and subroutines) from Fortran and C

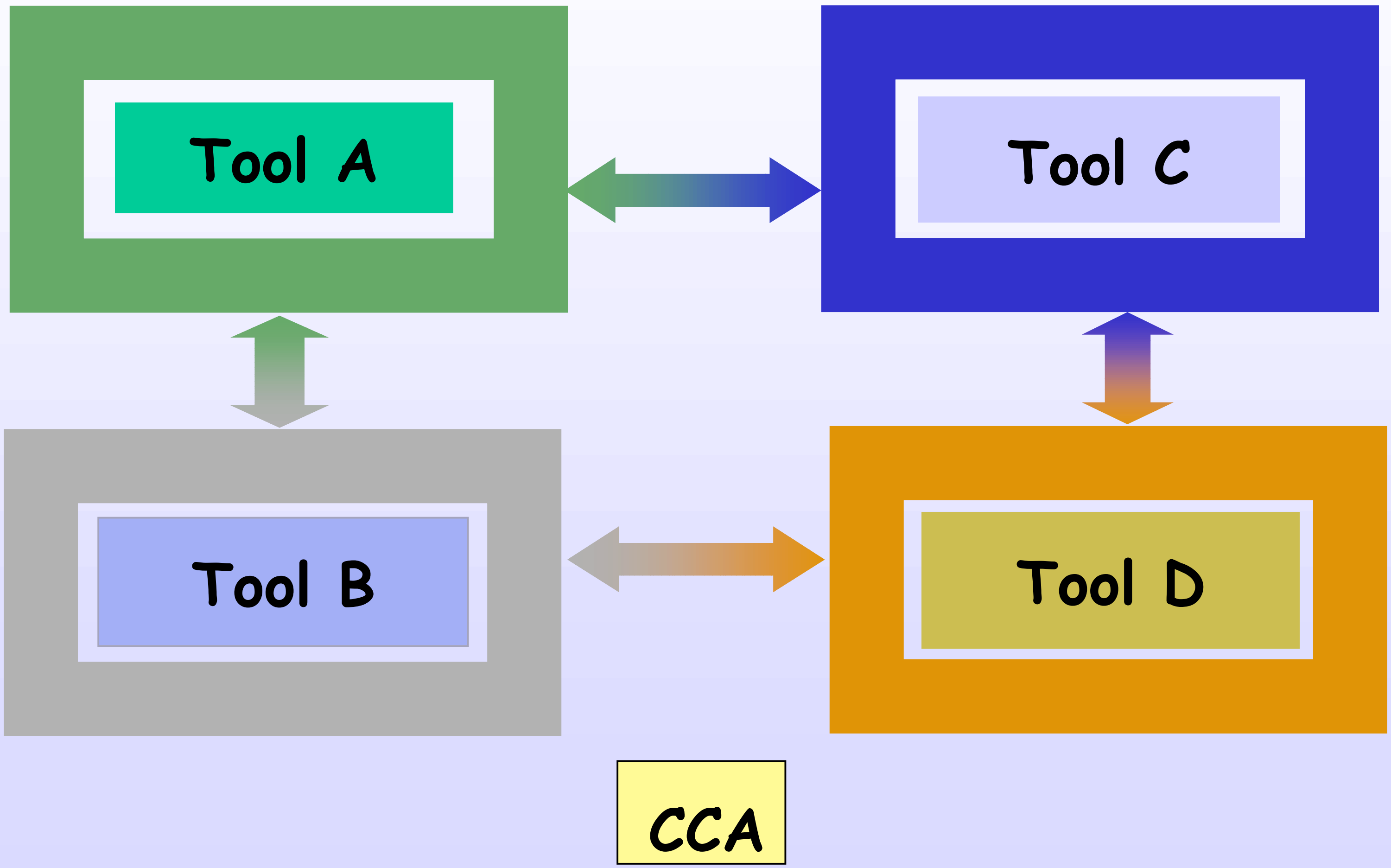
- Best approach is to start with examples for beginners!

- Several efforts are targeting Tool Interoperability!

Tool Interoperability
Tool-to-Tool



Component Technology!



PSE's and Frameworks

PMatlab

PyACTS

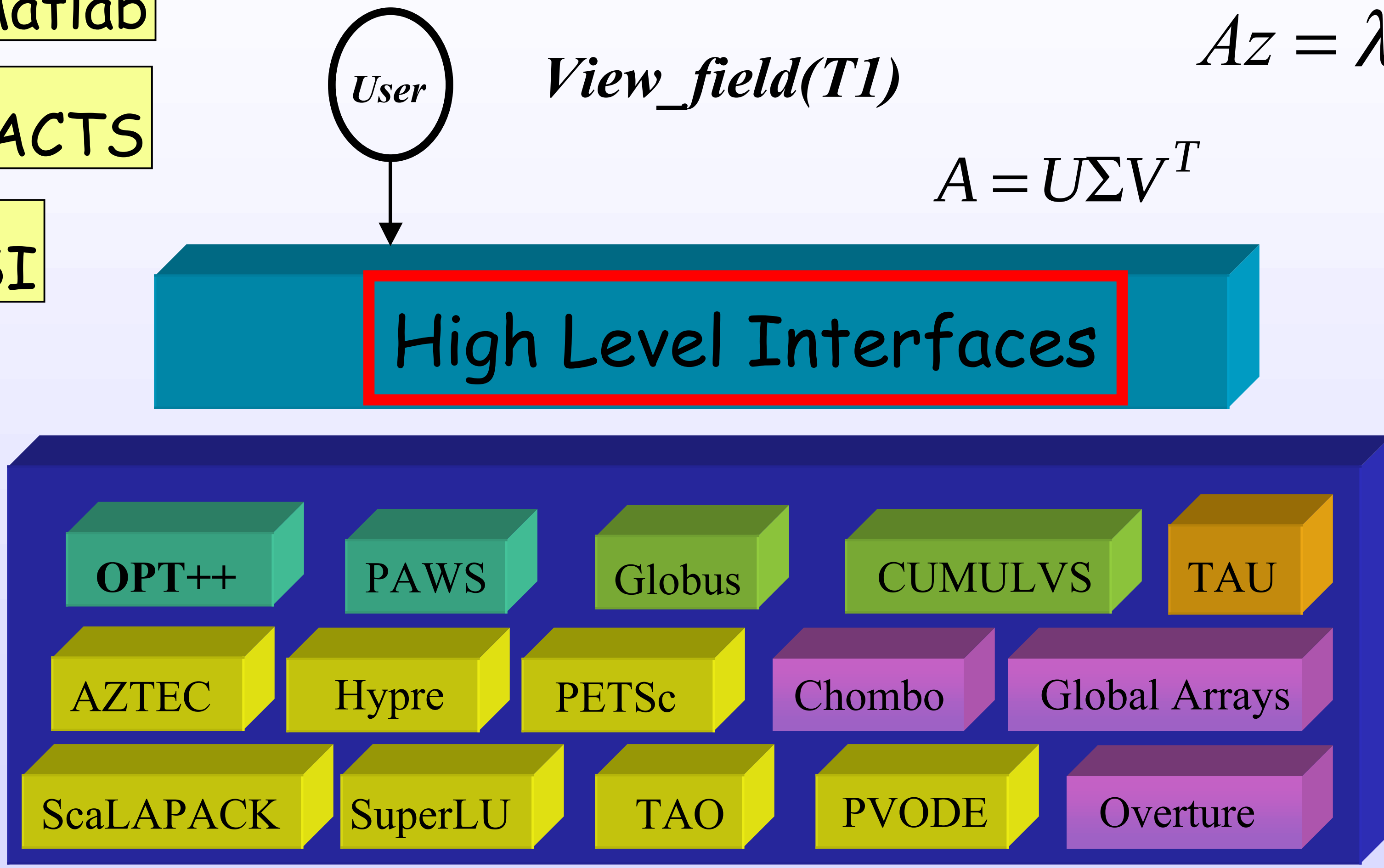
ESI

$$Ax = b$$

$$Az = \lambda z$$

View_field(T1)

$$A = U\Sigma V^T$$



This weeks agenda!

Agenda

<table>
<tr><th>Wednesday
Sep 4</th><th>Thursday
Sep 5</th><th>Friday
Sep 6</th><th>Saturday
Sep 7</th></tr>
<tr><td>Introduction to Computational Environments</td><td rowspan="2">Support for PDEs</td><td>Component Technology</td><td>Coupling</td></tr>
<tr><td rowspan="4">Direct Linear Solvers and Dense Eigenvalue Systems</td><td rowspan="2">Grid Manipulation</td><td rowspan="2">Performance And Tuning</td></tr>
<tr><td rowspan="3">Numerical Optimization</td></tr>
<tr><td>ODE</td><td rowspan="2">Collaborations and Industry</td></tr>
<tr><td>Remote Stearing and Visualization</td></tr>
</table>

acts-support@nersc.gov

http://acts.nersc.gov